

BAB II

KAJIAN PUSTAKA

2.1 Studi Sebelumnya

Penelitian-penelitian terdahulu menjadi landasan penting dalam pengembangan studi ini. Berbagai pendekatan telah digunakan oleh peneliti sebelumnya dalam melakukan prediksi harga rumah, termasuk algoritma *Regresi Linear*, *Random Forest*, dan *XGBoost*, baik secara individu maupun dikombinasikan melalui metode *ensemble learning* seperti *Stacking Regressor*. Hasil dari studi-studi tersebut dirangkum dalam Tabel 2.1 sebagai acuan perbandingan dan untuk memperkuat justifikasi pemilihan metode *Stacking Regressor* dalam penelitian ini.

Tabel 2.1 Studi Sebelumnya

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
1	(Nuris, 2024)	Analisis Prediksi Harga Rumah Pada <i>Machine learning</i> Menggunakan Metode <i>Regresi Linear</i>	Penelitian kuantitatif menggunakan algoritma <i>Regresi Linear</i> . Data sekunder diambil dari situs rumah123.com untuk wilayah Bekasi sebanyak 361 data. Tahapan meliputi: pengumpulan data, pra-pemrosesan (menghapus nilai hilang, menangani <i>outlier</i> , mengubah variabel	Model <i>Regresi Linear</i> yang dibangun menghasilkan akurasi prediksi sebesar 86,41% dengan nilai R^2 sebesar 0,7818811, MSE sebesar 464,87397, dan MAE sebesar 313,00083. Analisis korelasi menunjukkan bahwa variabel luas bangunan memiliki pengaruh paling besar terhadap harga

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
			kategorikal menjadi numerik), pembagian data menjadi 80% <i>training</i> dan 20% testing, pelatihan model, evaluasi menggunakan MSE, MAE, dan R^2, lalu prediksi pada data baru. 	rumah (nilai korelasi 0,80), sementara variabel jenis sertifikat tidak memberikan pengaruh signifikan (nilai korelasi 0,00). Penelitian ini menunjukkan bahwa faktor fisik properti, khususnya luas bangunan, merupakan penentu utama harga rumah di wilayah Bekasi. Selain itu, model yang dibangun mampu memprediksi harga rumah untuk properti yang belum pernah dianalisis sebelumnya, seperti pada contoh rumah dengan 3 kamar tidur, luas tanah 100 m², luas bangunan 70 m², sertifikat hak milik, daya listrik 2.200 VA, dan satu lantai, yang diprediksi bernilai sekitar 940 juta rupiah.
2	(Ade & Sheriffdeen,	Leveraging <i>XGBoost</i> ,	Penelitian ini menggunakan tiga model <i>machine learning</i> utama, yaitu <i>XGBoost</i> ,	Hasil penelitian menunjukkan bahwa <i>XGBoost</i> merupakan model dengan kinerja

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
	2024)	<i>Random Forest, and Linear Regression for Accurate Home Price Forecasting</i>	<i>Random Forest, dan Linear Regression, untuk memprediksi harga rumah berdasarkan atribut properti seperti lokasi, luas bangunan, dan fasilitas. Kinerja model dievaluasi dengan membandingkan tingkat akurasi dan ketahanannya terhadap data, sehingga dapat menilai kemampuan masing-masing algoritma dalam menangani hubungan non-linear dan interaksi antar variabel.</i>	terbaik dalam memprediksi harga rumah, unggul dalam hal akurasi dan ketahanan, khususnya pada <i>dataset</i> yang besar dan kompleks dengan banyak fitur. <i>Random Forest</i> memberikan hasil yang kompetitif dan mampu menangani hubungan non-linear serta interaksi antarvariabel dengan baik, meskipun sedikit berada di bawah <i>XGBoost</i> dari segi presisi. Sementara itu, <i>Linear Regression</i> berperan sebagai model <i>baseline</i> yang sederhana, mudah diinterpretasikan, dan efektif untuk data dengan hubungan linear yang kuat. Walaupun performanya sedikit lebih rendah pada data yang kompleks, model ini tetap memberikan gambaran awal yang jelas mengenai pengaruh masing-masing variabel terhadap harga rumah. Temuan ini

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
				menegaskan bahwa model berbasis pohon, seperti <i>XGBoost</i> dan <i>Random Forest</i> , umumnya lebih efektif dalam memodelkan interaksi variabel dan hubungan non-linear pada prediksi harga rumah.
3	(Sharma et al., 2024)	An Optimal House Price Prediction Algorithm: <i>XGBoost</i>	Penelitian ini menggunakan pendekatan regresi untuk memprediksi harga rumah dengan memanfaatkan <i>dataset</i> harga rumah dari Ames City, Iowa, USA. Beberapa algoritma <i>machine learning</i> diterapkan dan dibandingkan, yaitu <i>XGBoost</i> , Support Vector Regressor (SVR), <i>Random Forest</i> Regressor, Multilayer Perceptron (MLP), dan <i>Multiple Linear Regression</i> (MLR). Selain melakukan perbandingan kinerja model, penelitian ini juga	Hasil penelitian menunjukkan bahwa di antara berbagai model regresi yang diuji, <i>XGBoost</i> menjadi model dengan performa terbaik dalam prediksi harga rumah, dengan nilai akurasi yang sangat tinggi (misalnya, R-squared sebesar 0,997) dan <i>error</i> yang rendah. Selain itu, penerapan teknik optimasi <i>hyperparameter</i> seperti <i>tuning</i> parameter secara signifikan meningkatkan akurasi prediksi, sehingga disarankan untuk menggunakan model ini dengan parameter yang dioptimalkan. Model ini disebut

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
			mengidentifikasi faktor-faktor kunci yang mempengaruhi harga rumah berdasarkan hasil analisis dari model yang dibangun.	sebagai "Golden Model" karena keunggulannya dalam hasil prediksi.
4	(Hibatulloh et al., 2025)	Prediksi Harga Rumah di Bandung 2024 Menggunakan <i>Ensemble learning</i> : Analisis Komparatif dan Interpretabilitas	Penelitian ini memanfaatkan data harga rumah di Bandung hasil <i>web scraping</i> dari www.rumah123.com pada Maret 2024. Beberapa teknik <i>ensemble learning</i> dibandingkan, yakni <i>Random Forest</i> , <i>Gradient Boosting Machines</i> , <i>XGBoost</i> , <i>Linear Regression</i> , dan <i>Stacking Ensemble</i> , dengan evaluasi menggunakan MAE, RMSE, dan R-squared. Analisis interpretabilitas dilakukan dengan SHAP values untuk mengidentifikasi fitur paling berpengaruh terhadap harga rumah.	Hasil penelitian menunjukkan bahwa model <i>Stacking Ensemble</i> memperoleh kinerja terbaik dalam memprediksi harga rumah di Bandung tahun 2024, dengan nilai R^2 sebesar 0,9076, RMSE 0,311, dan MAE 0,216, terutama ketika fitur lokasi digunakan. Analisis interpretabilitas menggunakan teknik SHAP mengungkapkan bahwa fitur ukuran tanah, ukuran bangunan, dan lokasi memiliki dampak terbesar terhadap prediksi harga. Selain itu, model ini berhasil diintegrasikan ke dalam situs web Flask untuk prediksi harga secara interaktif.

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
5	(Qolbi et al., 2025)	Prediksi Harga Rumah Di Jakarta Pusat Menggunakan Algoritma <i>Machine learning</i>	Penelitian ini menggunakan metode pengumpulan data melalui teknik <i>web scraping</i> dengan cakupan lokasi di Jakarta Pusat. Data yang diperoleh kemudian diolah dan digunakan untuk membangun model prediksi harga rumah menggunakan tiga algoritma, yaitu <i>Regresi Linear</i> , <i>Support Vector Machine</i> (SVM), dan <i>Random Forest</i> (RF). Kinerja masing-masing model dievaluasi menggunakan beberapa metrik, antara lain <i>Mean Absolute Error</i> (MAE), <i>Root Mean Squared Error</i> (RMSE), koefisien determinasi (R^2), dan <i>Mean Absolute Percentage Error</i> (MAPE) untuk menentukan algoritma dengan performa terbaik.	Hasil penelitian menunjukkan bahwa dari ketiga algoritma <i>machine learning</i> yang digunakan, <i>Random Forest</i> memberikan performa terbaik dalam memprediksi harga rumah di Jakarta Pusat. Nilai MAE sebesar 3023, RMSE sebesar 7112, R^2 sebesar 0,942, dan MAPE sebesar 0,269 menunjukkan akurasi yang cukup tinggi. Selain itu, variabel yang paling berpengaruh terhadap harga rumah adalah luas tanah dan luas bangunan.

No	Penulis & Tahun	Judul Penelitian	Metode	Hasil Temuan
6	(Fauzi et al., 2025)	Prediksi Harga Properti Di Indonesia Menggunakan Algoritma <i>Random Forest</i>	Penelitian ini bertujuan untuk mengembangkan model prediksi harga properti di Indonesia menggunakan algoritma <i>Random Forest</i> , menganalisis faktor utama yang mempengaruhi harga, dan mengevaluasi kinerja model dengan metrik MAE dan RMSE. Metode yang digunakan meliputi pengumpulan <i>dataset</i> harga properti yang terdiri dari 10.000 sampel, preprocessing data, pembangunan model menggunakan algoritma <i>Random Forest</i> , serta evaluasi model dengan metrik MAE dan RMSE.	Hasil penelitian menunjukkan bahwa model <i>Random Forest</i> mampu memprediksi harga properti dengan tingkat akurasi yang cukup baik, dengan nilai MAE sebesar Rp2,48 miliar dan RMSE sebesar Rp2,89 miliar. Faktor utama yang mempengaruhi harga adalah jarak ke pusat kota (22,2%), diikuti oleh luas bangunan (20,1%), luas tanah (18,5%), tahun dibangun (15,0%), dan jumlah kamar tidur (6,7%). Visualisasi perbandingan harga aktual dan prediksi menunjukkan sebagian besar prediksi mendekati nilai aktual, meskipun terdapat beberapa <i>outlier</i> .

Sejumlah penelitian terdahulu telah mengeksplorasi prediksi harga rumah menggunakan berbagai algoritma *machine learning*, baik secara terpisah maupun melalui pendekatan *ensemble*. *XGBoost* secara berulang menunjukkan kinerja prediktif yang menonjol dibandingkan *Random Forest* dan *Linear Regression*, khususnya ketika dihadapkan pada *dataset* berukuran besar dengan banyak jenis variabel atau atribut (kaya fitur), sehingga mampu mempertahankan akurasi yang tinggi. *Linear Regression*, meskipun unggul dari sisi interpretabilitas, cenderung memberikan akurasi lebih rendah pada data yang kompleks. Salah satu penelitian melaporkan bahwa penerapan *hyperparameter tuning* pada *XGBoost* mampu mendorong nilai R^2 hingga mencapai 0,997, menjadikannya sebagai model unggulan (*golden model*). Sementara itu, *Random Forest* juga terbukti efektif dalam menangkap hubungan non-linear antar variabel, seperti yang terlihat pada studi di Jakarta Pusat dengan R^2 sebesar 0,942. Faktor-faktor yang sering diidentifikasi sebagai penentu utama harga rumah meliputi luas tanah, luas bangunan, dan jarak ke pusat kota.

Pendekatan *ensemble learning* lain, seperti *Stacking Regressor*, juga menunjukkan performa yang kompetitif dalam meningkatkan akurasi prediksi. Penelitian di Bandung mengungkap bahwa *Stacking* memberikan hasil terbaik dengan nilai R^2 sebesar 0,9076, terutama ketika fitur lokasi disertakan dalam model. Analisis interpretabilitas berbasis SHAP mengindikasikan bahwa ukuran tanah, ukuran bangunan, dan lokasi merupakan variabel yang paling mempengaruhi hasil prediksi. Keunggulan lainnya adalah keberhasilan integrasi model ini ke dalam aplikasi web interaktif. Berdasarkan rangkaian bukti tersebut, penggunaan metode *ensemble* seperti *Stacking Regressor* dinilai relevan untuk penelitian ini dalam rangka meningkatkan akurasi prediksi harga rumah tinggal di wilayah perkotaan Indonesia, khususnya Depok, Bandung, dan Jakarta Selatan.

2.2 Dasar Teori

2.2.1 Platform *Kaggle* sebagai Sumber *Dataset* Publik

Kaggle merupakan platform berbasis komunitas yang dikembangkan oleh *Google* dan secara luas dikenal sebagai salah satu ekosistem ilmu data (*data science*) terbesar di dunia. Platform ini menyediakan berbagai layanan yang mendukung kegiatan penelitian dan pengembangan model *machine learning*, di antaranya kompetisi ilmu data, forum diskusi, lingkungan komputasi berbasis *cloud* (*notebook*), serta repositori *dataset* publik yang dapat diakses secara bebas oleh siapa pun. Hingga saat ini, *Kaggle* telah memiliki jutaan pengguna aktif dari kalangan akademisi, praktisi, maupun peneliti di seluruh dunia, menjadikannya sebagai salah satu referensi utama dalam ekosistem riset berbasis data.

Salah satu fitur utama *Kaggle* yang paling banyak dimanfaatkan dalam penelitian adalah repositori *dataset* publiknya. *Dataset* yang tersedia di *Kaggle* umumnya telah diunggah dalam format tabular yang terstruktur seperti CSV, sehingga dapat langsung digunakan tanpa memerlukan proses ekstraksi data yang kompleks. Sebagai contoh, *dataset House Prices – Advanced Regression Techniques* yang tersedia di *Kaggle* memuat sekitar seratus variabel atribut rumah, seperti luas lahan, lokasi, luas bangunan, jumlah kamar mandi, dan jumlah kamar tidur, dalam format yang siap digunakan untuk pelatihan model (B, 2023; Li, 2023). Selain itu, banyak *dataset* di *Kaggle* telah melalui proses kurasi awal oleh pengunggahnya, sehingga kualitas datanya relatif terjamin. Penggunaan *dataset* publik dari *Kaggle* juga menghindari permasalahan legalitas yang kerap muncul pada metode pengumpulan data secara mandiri, seperti pelanggaran *Terms of Service* suatu situs web.

Dalam konteks penelitian *machine learning*, *Kaggle* telah banyak digunakan sebagai sumber *dataset* oleh berbagai penelitian ilmiah. Beberapa penelitian di bidang prediksi harga rumah secara khusus menggunakan *dataset* dari *Kaggle* sebagai data utama karena kemudahan aksesnya, ketersediaan dokumentasi yang jelas, serta sifatnya yang dapat

direproduksi oleh peneliti lain (B, 2023; Li, 2023; Liao, 2023). (Liao, 2023) misalnya menggunakan *dataset* harga rumah *Seattle* yang bersumber dari *Kaggle*, yang memuat data 2.515 rumah tinggal dengan delapan fitur terstruktur, untuk membandingkan delapan model *machine learning* secara komprehensif. Hal ini menjadikan *Kaggle* sebagai pilihan yang tepat untuk penelitian yang mengutamakan transparansi dan keterbukaan data.

2.2.2 Jenis Properti: Rumah Tinggal

Jenis properti yang menjadi fokus dalam penelitian ini adalah rumah tinggal. Rumah tinggal dipilih karena merupakan bentuk properti yang paling umum dan mendasar dalam memenuhi kebutuhan papan masyarakat. Tidak seperti properti lain seperti vila, rumah kos, atau apartemen yang cenderung bersifat komersial atau memiliki segmen pasar khusus, rumah tinggal berfungsi sebagai tempat tinggal permanen bagi individu dan keluarga dari berbagai latar belakang sosial dan ekonomi. Hal ini menjadikannya sebagai objek yang lebih representatif dalam mencerminkan kondisi pasar properti residensial secara umum (Saputra & Pratama, 2025).

Pemisahan jenis properti ini juga dilakukan demi menjaga validitas dan konsistensi dalam analisis data, khususnya dalam hal prediksi harga. Setiap jenis properti memiliki karakteristik fisik, segmentasi pasar, serta faktor penentu harga yang berbeda. Misalnya, harga apartemen sangat dipengaruhi oleh fasilitas gedung dan tingkat lantai, sementara rumah kos cenderung dipengaruhi oleh jumlah kamar sewa dan potensi pemasukan bulanan. Oleh karena itu, mencampur jenis properti dalam satu model prediktif akan menghasilkan variabilitas data yang tinggi dan menurunkan keakuratan hasil prediksi (Fitri, 2023). Dengan memusatkan analisis hanya pada rumah tinggal, model yang digunakan dapat mengidentifikasi pola harga dengan lebih tepat dan menghasilkan prediksi yang lebih seragam dan dapat diandalkan.

Secara sosial, rumah tinggal memiliki peran penting dalam kehidupan masyarakat karena menyangkut kebutuhan dasar akan tempat berlindung, keamanan, dan kenyamanan. Permintaan terhadap rumah tinggal terus

meningkat seiring pertumbuhan penduduk, urbanisasi, serta meningkatnya kesadaran akan pentingnya memiliki hunian yang layak. Bahkan data nasional menunjukkan bahwa backlog atau kekurangan rumah masih menjadi tantangan besar yang dihadapi pemerintah, dengan jutaan keluarga belum memiliki tempat tinggal tetap (Aufa et al., 2022). Oleh karena itu, rumah tinggal tidak hanya relevan sebagai objek analisis ekonomi, tetapi juga sebagai isu strategis dalam kebijakan sosial dan pembangunan perumahan nasional.

2.2.3 *Data Mining*

Data mining merupakan proses komputasi yang bertujuan untuk mengekstraksi informasi penting, pola tersembunyi, serta pengetahuan baru dari kumpulan data dalam jumlah besar. Proses ini menggabungkan berbagai teknik dari disiplin ilmu seperti kecerdasan buatan, pembelajaran mesin, statistika, dan sistem basis data. Dalam dunia akademik dan industri, data mining sering kali disandingkan dengan istilah *knowledge discovery in databases* (KDD), yang mengacu pada keseluruhan tahapan dalam menemukan wawasan bermakna dari data yang tidak selalu terstruktur atau bersih (Wu et al., 2021). Layaknya proses menambang logam mulia dari perut bumi, data mining berupaya menyaring "emas informasi" dari kumpulan data mentah yang kompleks dan beragam (Gheisari et al., 2023).

Lebih dari sekadar eksplorasi data, tujuan utama data mining adalah menemukan pola atau struktur yang tersembunyi dan menggunakannya untuk membangun model yang mampu menjelaskan fenomena atau memprediksi kejadian di masa mendatang. Manfaatnya pun sangat luas, mulai dari membantu perusahaan memahami perilaku konsumen hingga mendukung pengambilan keputusan dalam sistem kesehatan. Pada sektor properti, misalnya, data mining dimanfaatkan untuk menganalisis keterkaitan antara karakteristik rumah seperti luas tanah, jumlah kamar, dan lokasi dengan harga jualnya. Hal ini tercermin dalam penelitian yang menggunakan algoritma *Random Forest* untuk memprediksi harga rumah di Surabaya, yang menunjukkan bahwa pendekatan berbasis data mining dapat menghasilkan estimasi yang lebih akurat dan informatif bagi para pemangku

kepentingan (Wardani et al., 2024). Dengan demikian, data mining tidak hanya bersifat deskriptif, tetapi juga prediktif, serta adaptif terhadap kebutuhan di berbagai bidang.

Dalam implementasinya, banyak teknik dalam *data mining* yang berasal dari cabang ilmu *machine learning*, sehingga keduanya memiliki keterkaitan yang sangat erat. *Machine learning* berperan sebagai fondasi algoritmik yang memungkinkan sistem mempelajari pola dari data secara otomatis tanpa diprogram secara eksplisit. Berbagai algoritma seperti *decision tree*, *support vector machine*, *Random Forest*, hingga *neural networks* merupakan contoh metode pembelajaran mesin yang banyak digunakan dalam aktivitas *data mining* (Gheisari et al., 2023). Oleh karena itu, dapat dikatakan bahwa *machine learning* merupakan bagian integral dari data mining, terutama dalam konteks pengembangan model prediktif. Seiring dengan meningkatnya kompleksitas dan volume data di era digital, kolaborasi antara *data mining* dan *machine learning* menjadi semakin penting dalam mendorong inovasi, efisiensi, dan ketepatan dalam pengambilan keputusan berbasis data (Wu et al., 2021).

2.2.4 *Machine Learning*

Machine learning merupakan salah satu cabang dari kecerdasan buatan (*artificial intelligence*) yang bertujuan mengembangkan sistem yang mampu belajar dari data secara otomatis tanpa perlu diprogram secara eksplisit. Dengan memanfaatkan algoritma yang dapat mengidentifikasi pola dalam data historis, *machine learning* memungkinkan proses prediksi atau pengambilan keputusan terhadap data baru secara efisien. Dalam konteks prediksi harga rumah, pendekatan ini memungkinkan model mempelajari hubungan antara fitur-fitur seperti luas bangunan, jumlah kamar, lokasi, dan fasilitas, dengan harga properti (Haryanto et al., 2023).

Pendekatan *machine learning* secara umum terbagi menjadi dua kategori utama, yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* digunakan ketika data pelatihan telah memiliki label atau target *output* yang jelas. Metode ini sering digunakan untuk prediksi

harga rumah, karena target yang diprediksi berbentuk nilai kontinu. Sebaliknya, *unsupervised learning* digunakan pada data tanpa label untuk menemukan pola atau struktur tersembunyi, misalnya dalam pengelompokan properti berdasarkan kesamaan karakteristik (Saiful et al., 2021).

Dalam supervised learning, dua jenis pendekatan yang umum digunakan adalah regresi dan klasifikasi. Regresi digunakan ketika target *output* berupa nilai numerik yang kontinu, sedangkan klasifikasi digunakan untuk memetakan input ke dalam kelas-kelas diskrit. Studi oleh (Haryanto et al., 2023) menunjukkan bahwa algoritma *Random Forest Regression* mampu menghasilkan prediksi harga rumah yang lebih akurat dibandingkan *Linear Regression*, karena kemampuannya menangani data nonlinier dan interaksi antar fitur.

2.2.5 Regresi dalam *Machine Learning*

Regresi merupakan salah satu metode dasar dalam statistik dan *machine learning* yang berfungsi untuk memodelkan hubungan antara satu atau lebih variabel independen dengan variabel dependen yang bersifat kontinu. Dalam konteks *machine learning*, regresi termasuk dalam kelompok algoritma *supervised learning* yang digunakan untuk memprediksi nilai numerik berdasarkan data historis (Sembiring & Sembiring, 2024). Teknik ini menjadi krusial ketika tujuan analisis adalah menghasilkan estimasi kuantitatif yang akurat, seperti prediksi pendapatan, harga, konsumsi energi, atau jumlah produksi.

Dalam implementasinya, regresi tidak hanya digunakan untuk menganalisis keterkaitan antarvariabel, tetapi juga sebagai alat pendukung pengambilan keputusan strategis berbasis data. Penelitian oleh (Suparmadi & Ramadhani, 2022) menunjukkan bagaimana regresi dapat diintegrasikan dalam sistem berbasis *machine learning* untuk memperkirakan pencapaian target profit perusahaan. Studi tersebut memanfaatkan data laporan laba/rugi dari sebuah perusahaan distributor untuk melatih beberapa model regresi, seperti *Linear Regression*, *Support Vector Regression (SVR)*,

Decision Tree Regressor, *Random Forest*, hingga *Gradient Boosting*. Setelah dilakukan evaluasi performa, model dengan tingkat kesalahan terendah dipilih dan di-*deploy* dalam sistem berbasis web menggunakan Python dan HTML. Sistem ini bertujuan untuk memberikan estimasi profit secara cepat dan akurat guna mendukung evaluasi kinerja manajerial yang lebih objektif dan terukur (Suparmadi & Ramadhani, 2022).

Senada dengan itu, (Sembiring & Sembiring, 2024) mengkaji kinerja berbagai model regresi dalam memprediksi harga beras, sebuah komoditas strategis yang fluktuasinya sangat berpengaruh terhadap daya beli masyarakat. Dalam studi tersebut, tujuh model regresi diuji, yaitu *Linear Regression*, *Linear Support Vector Regression*, *RBF Support Vector Regression*, *Decision Tree Regression*, *Random Forest Regression*, *Gradient Boosting Regression*, dan *MLP Regressor*. Evaluasi dilakukan dengan menggunakan metrik performa seperti *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, serta *R²-score*. Hasilnya menunjukkan bahwa *Decision Tree Regression* secara konsisten memberikan hasil estimasi terbaik, dengan nilai kesalahan mendekati nol dan skor *R²* mencapai 1,00 dalam berbagai rasio pengujian data (90:10 hingga 60:40). Temuan ini menegaskan bahwa pemilihan model regresi yang tepat sangat menentukan akurasi prediksi dan efektivitas implementasi sistem prediktif di dunia nyata (Sembiring & Sembiring, 2024).

2.2.6 Ensemble Learning

Ensemble learning adalah pendekatan dalam *machine learning* yang menggabungkan beberapa model dasar (*base learners*) untuk memperoleh prediksi yang lebih akurat, stabil, dan andal dibandingkan dengan model tunggal. Teknik ini memanfaatkan prinsip bahwa kombinasi dari banyak model sederhana dapat menghasilkan kinerja yang lebih baik daripada satu model kompleks yang berdiri sendiri (Hibatulloh et al., 2025).

Tiga pendekatan utama dalam *ensemble learning* adalah *bagging*, *boosting*, dan *stacking*. *Bagging* (misalnya *Random Forest*) fokus pada

pengurangan variansi melalui pelatihan paralel dengan data subset acak. *Boosting* (seperti *XGBoost*) fokus pada pengurangan bias melalui pelatihan bertahap. Sementara *Stacking* menggabungkan beberapa model melalui *meta learner* untuk mempelajari kombinasi optimal dari hasil-hasil prediksi.

Keunggulan utama dari *ensemble learning* terletak pada kemampuannya untuk mengurangi *overfitting*, meningkatkan generalisasi, dan memanfaatkan keunikan dari setiap model dasar. Oleh karena itu, teknik ini menjadi bagian penting dalam pengembangan sistem prediktif modern di berbagai bidang, termasuk prediksi harga rumah.

2.2.7 *Data cleaning*

Data cleaning adalah proses penting dalam tahap awal pengolahan data yang bertujuan untuk menghilangkan noise, memperbaiki inkonsistensi, dan menyaring data yang tidak relevan agar *dataset* menjadi bersih, terstruktur, dan siap digunakan dalam proses analitik lebih lanjut. Tahap ini menjadi fondasi penting dalam *data science* dan *machine learning* karena kualitas data sangat memengaruhi akurasi dan ketepatan hasil prediksi model (Agung et al., 2023; Soares et al., 2025).

Dalam konteks studi prediksi harga rumah, proses *Data cleaning* dilakukan secara sistematis untuk memastikan hanya informasi yang relevan dan valid yang digunakan dalam pelatihan model. Berikut adalah tahapan-tahapan utama dalam proses *Data cleaning*:

1. Penghapusan Data Duplikat

Penghapusan data duplikat merupakan langkah penting untuk memastikan integritas *dataset*. Data yang identik pada baris berbeda dapat menyebabkan perhitungan ganda dan bias pada hasil analisis. Proses ini melibatkan identifikasi dan penghapusan entri yang sama secara persis, baik dari segi nilai maupun struktur, untuk meningkatkan akurasi model dan mengurangi redundansi (Agung et al., 2023).

2. Penanganan *Missing values*

Missing values atau nilai yang hilang merupakan masalah umum dalam *dataset* real-world yang dapat mempengaruhi kualitas analisis dan performa model machine learning (Emmanuel et al., 2021). Oleh karena itu, penanganan *missing values* perlu dilakukan sebagai bagian dari tahap *data cleaning* untuk memastikan kelengkapan data sebelum proses pemodelan dimulai. Apabila tidak ditemukan missing value, *dataset* dapat langsung dilanjutkan ke tahap berikutnya tanpa memerlukan tindakan lebih lanjut (Emmanuel et al., 2021).

3. Validasi Logis terhadap Nilai Atribut

Validasi logis dilakukan untuk mendeteksi nilai-nilai yang tidak masuk akal atau melanggar aturan domain, seperti harga, luas tanah, atau luas bangunan yang bernilai nol atau negatif, maupun jumlah kamar tidur dan kamar mandi yang berada di luar batas wajar untuk hunian tempat tinggal. Entri semacam ini dapat mencerminkan kesalahan input atau noise dalam data, dan perlu dihapus agar tidak mempengaruhi performa model analitik (Agung et al., 2023).

4. Transformasi Logaritmik pada Variabel Target

Data harga properti pada praktiknya cenderung memiliki distribusi yang sangat menceng ke kanan (*right-skewed*), karena mayoritas properti berada pada kisaran harga menengah sementara sebagian kecil properti premium bernilai jauh lebih tinggi dari populasi utamanya. Kondisi ini dapat menyulitkan metode deteksi *outlier* berbasis sebaran statistik seperti IQR, karena batas atas yang dihasilkan cenderung terlalu lebar akibat ekor distribusi yang panjang, sehingga properti bernilai tinggi namun sebenarnya masih wajar berpotensi ikut terhapus, atau sebaliknya *outlier* yang sesungguhnya tidak terdeteksi secara akurat. Salah satu pendekatan umum untuk mengatasi hal ini adalah menerapkan transformasi logaritmik pada variabel target sebelum proses deteksi *outlier* maupun pelatihan model dilakukan, sehingga distribusi data menjadi lebih simetris dan mendekati distribusi normal. (Khan et al., 2026) menerapkan pendekatan ini pada penelitian prediksi harga rumah menggunakan *Stacking Ensemble* dengan *base learner* berbasis pohon (*XGBoost*, *CatBoost*, *LightGBM*) dan *Ridge*

Regression sebagai *meta learner*, di mana variabel target ditransformasi secara logaritmik untuk mengurangi skewness sebelum pelatihan model dilakukan, kemudian hasil prediksi dikembalikan ke skala aslinya (inverse transform) sebelum evaluasi akhir dihitung. Pendekatan ini terbukti efektif, dengan model *Stacking* yang dihasilkan mencapai performa terbaik dibandingkan model tunggal maupun pendekatan weighted averaging.

5. Deteksi dan Penghapusan *Outlier* dengan IQR

Outlier dapat berdampak signifikan terhadap hasil analisis, terutama pada algoritma yang sensitif terhadap distribusi data seperti regresi linear. Metode *Interquartile Range (IQR)* digunakan untuk mengidentifikasi nilai ekstrem di luar rentang $Q1 - 1.5 \times IQR$ hingga $Q3 + 1.5 \times IQR$. Nilai-nilai tersebut kemudian dihapus untuk meningkatkan stabilitas dan akurasi model (Soares et al., 2025). Selain diterapkan sebagai tahap pembersihan data awal, *IQR* juga dapat diintegrasikan ke dalam *pipeline* pemodelan, di mana batas bawah dan batas atas dihitung secara eksklusif dari data pelatihan dan kemudian diterapkan secara independen ke seluruh subset data. Pendekatan ini penting untuk mencegah *data leakage*, yaitu kondisi di mana informasi dari data uji ikut memengaruhi proses preprocessing sehingga menghasilkan estimasi performa model yang tidak realistis (Prusty et al., 2022). Namun demikian, pendekatan berbasis distribusi statistik seperti *IQR* memiliki keterbatasan pada variabel diskrit, di mana batas yang dihasilkan dapat bernilai tidak logis akibat kemiringan distribusi data (Çilgin et al., 2023). Sebagai alternatif, penanganan *outlier* pada variabel diskrit dapat dilakukan menggunakan pendekatan *domain knowledge*, yaitu penetapan batas nilai berdasarkan kelaziman logis dalam domain properti hunian, bukan berdasarkan sebaran statistik data (Çilgin et al., 2023).

2.2.8 *Feature engineering*

Feature engineering adalah proses menciptakan fitur atau variabel baru dari fitur yang sudah ada untuk meningkatkan performa model *machine learning* (Verdonck et al., 2024). Dalam konteks prediksi harga properti, fitur turunan dapat memberikan insight tambahan yang tidak tersedia secara langsung dari data mentah. Fitur-fitur turunan ini seringkali memiliki

korelasi yang lebih kuat dengan target prediksi dibandingkan fitur asli, sehingga dapat meningkatkan akurasi model (Deng, 2025; Verdonck et al., 2024).

Dalam domain prediksi harga properti, beberapa fitur turunan memiliki landasan teoritis yang kuat. (Diewert & Shimizu, 2022) menunjukkan bahwa normalisasi nilai properti terhadap luas, seperti harga per m² tanah dan harga per m² bangunan, merupakan representasi yang lebih informatif dibandingkan nilai absolut. Selain itu, rasio luas bangunan terhadap luas tanah terbukti lebih stabil secara statistik dibandingkan menggunakan kedua variabel secara terpisah, mengingat potensi multikolinearitas di antara keduanya (Diewert & Shimizu, 2022). Terkait fitur berbasis jumlah ruangan, (Hibatulloh et al., 2025) membuktikan melalui analisis SHAP bahwa atribut-atribut berbasis jumlah ruangan seperti `bedroom_count` dan `bathroom_count` masing-masing berpengaruh positif terhadap harga properti, sehingga keduanya dapat dikategorikan sebagai fitur yang relevan untuk merepresentasikan kapasitas hunian suatu properti.

2.2.9 *Exploratory Data Analysis / EDA*

Eksplorasi Data atau *Exploratory Data Analysis* (EDA) merupakan tahap penting dalam proses analisis data yang bertujuan untuk memahami karakteristik dan struktur dasar dari *dataset*. Tahapan ini dilakukan sebelum membangun model prediktif, dengan tujuan untuk mengidentifikasi pola, mendeteksi anomali, menguji asumsi, serta mengevaluasi distribusi dan hubungan antar variabel (Dhany et al., 2023).

EDA memberikan pemahaman awal yang kuat terhadap data yang digunakan, sehingga analisis lanjutan dapat dilakukan dengan pendekatan yang lebih tepat dan terarah. Selain itu, EDA membantu peneliti menentukan strategi pembersihan dan transformasi data, serta mendukung dalam proses seleksi fitur yang relevan.

Dalam praktiknya, EDA menggunakan berbagai teknik, di antaranya:

- **Statistik Deskriptif:**

Teknik ini digunakan untuk menyajikan ringkasan data numerik, seperti nilai minimum, maksimum, rata-rata (*mean*), median, modus, standar deviasi, dan kuartil. Statistik deskriptif memberikan gambaran umum terhadap skala dan persebaran data, serta membantu mengidentifikasi potensi *outlier* dan *skewness*.

- **Visualisasi Distribusi:**

Visualisasi seperti *histogram*, *boxplot*, dan *density plot* digunakan untuk memahami distribusi nilai dari masing-masing fitur. Teknik ini efektif untuk mendeteksi anomali, *outlier*, serta perbedaan pola distribusi antar variabel.

- **Korelasi antar Fitur:**

Analisis korelasi bertujuan untuk mengukur hubungan antar variabel numerik. Matriks korelasi atau heatmap korelasi sering digunakan untuk mengidentifikasi kekuatan hubungan antara fitur prediktor dengan variabel target, sehingga dapat menjadi dasar pertimbangan dalam proses seleksi fitur yang relevan terhadap target prediksi.

2.2.10 Feature Selection

Seleksi fitur merupakan proses pemilihan subset fitur yang paling relevan dari keseluruhan fitur yang tersedia dalam *dataset* untuk digunakan dalam pembangunan model *machine learning*. Proses ini menjadi penting karena tidak semua fitur berkontribusi positif terhadap performa model, di mana keberadaan fitur yang tidak relevan justru dapat menambah *noise* dan menurunkan akurasi prediksi. (Akyap, 2023) menegaskan bahwa seleksi fitur merupakan komponen penting dalam *machine learning*, terutama ketika peneliti dihadapkan pada data berdimensi tinggi yang apabila tidak ditangani dengan tepat dapat berdampak negatif terhadap performa model secara keseluruhan. Secara umum, metode seleksi fitur diklasifikasikan ke dalam tiga kategori utama, yaitu metode *filter*, *wrapper*, dan *embedded*, di mana masing-masing kategori memiliki pendekatan yang berbeda dalam

mengevaluasi dan memilih fitur yang relevan terhadap variabel target (Akyap, 2023).

2.2.11 Data Preprocessing

Pra-pemrosesan data merupakan tahap krusial dalam *pipeline* pembelajaran mesin untuk memastikan kualitas dan kesesuaian data sebelum digunakan dalam pelatihan model. Tahapan ini mencakup berbagai teknik seperti *encoding* variabel kategorikal, normalisasi atau standarisasi data numerik, serta seleksi fitur untuk meningkatkan efisiensi model. Selain itu, perlu pula dibedakan antara pra-pemrosesan dengan *Data cleaning*, yang kerap dianggap serupa namun memiliki fokus yang berbeda.

1. Encoding Variabel Kategorikal

Encoding variabel kategorikal dilakukan untuk mengubah nilai non-numerik menjadi bentuk numerik yang dapat diproses oleh algoritma. Beberapa pendekatan yang umum digunakan antara lain label *encoding* yang mengasumsikan adanya urutan antar nilai kategori, ordinal *encoding* yang digunakan untuk variabel kategorikal yang memiliki tingkatan tertentu, serta mean *encoding* yang menggantikan setiap kategori dengan rata-rata nilai target yang bersesuaian. Namun, mean *encoding* biasa rentan terhadap *overfitting* pada kategori dengan sedikit sampel, sehingga dikembangkan varian yang lebih stabil yaitu Smoothed Target *Encoding*. Metode ini bekerja dengan menggabungkan dua sumber informasi: rata-rata nilai target pada kategori tertentu dan rata-rata nilai target secara global, dengan proporsi yang ditentukan oleh jumlah sampel pada kategori tersebut. Kategori yang memiliki banyak sampel akan lebih banyak dipengaruhi oleh rata-rata kategorinya sendiri, sedangkan kategori dengan sedikit sampel akan ditarik mendekati rata-rata global sehingga estimasinya lebih konservatif dan stabil. Dengan cara ini, Smoothed Target *Encoding* mampu menghindari *overfitting* pada kategori yang jarang muncul dalam data pelatihan, sekaligus tetap mempertahankan informasi spesifik dari kategori yang memiliki sampel yang cukup banyak (Pargent et al., 2022).

2. Pembagian Data (*Train-Test Split*)

Pembagian data (*data splitting*) merupakan proses membagi *dataset* menjadi *training set* dan *testing set* dengan tujuan memisahkan proses pembelajaran model dari proses evaluasinya, sehingga model mampu melakukan generalisasi terhadap data baru yang belum pernah dilihat sebelumnya. Salah satu teknik yang umum digunakan adalah *random split*, yaitu pemilihan data secara acak untuk menjaga distribusi yang seimbang di kedua subset dan menghindari potensi bias (Hibatulloh et al., 2025). Apabila *dataset* mengandung variabel kategorikal dengan distribusi kelas yang tidak merata, *stratified split* dapat digunakan sebagai alternatif untuk memastikan proporsi setiap kategori terjaga secara merata di seluruh subset data. Teknik ini membagi data berdasarkan nilai variabel tertentu sehingga setiap subset merepresentasikan distribusi keseluruhan *dataset* secara proporsional, yang pada akhirnya menghasilkan evaluasi model yang lebih andal dan tidak bias terhadap kategori tertentu (Prusty et al., 2022). Rasio pembagian yang banyak direkomendasikan dalam studi regresi adalah 80:20 atau 70:30 (Nguyen et al., 2021). Penelitian ini mengadopsi rasio 70:30, di mana 70% data digunakan sebagai data *training* dan 30% sisanya disisihkan sebagai data test untuk evaluasi akhir. Pada pendekatan *Stacking* dengan OOF, *meta learner* dilatih menggunakan prediksi OOF yang dihasilkan dari *K-Fold cross-validation* pada data *training*, sehingga tidak diperlukan set data terpisah ketiga khusus untuk melatih *meta learner* (Park et al., 2022). Pemisahan yang ketat antara data *training* dan data test tetap penting untuk mencegah data leakage yang dapat menyebabkan estimasi performa model menjadi terlalu optimistis (Mienye et al., 2022). Pemisahan ini penting untuk mencegah *data leakage* antar tahap pelatihan yang dapat menyebabkan estimasi performa model menjadi terlalu optimistis. Selain itu, penetapan *random state* yang tetap perlu dilakukan untuk menjaga *reproducibility* hasil evaluasi (Hibatulloh et al., 2025).

3. Standarisasi Data

Standarisasi data merupakan teknik transformasi yang digunakan untuk menyelaraskan skala antar fitur numerik, sehingga mendukung efektivitas

algoritma yang sensitif terhadap perbedaan skala seperti regresi linear. Salah satu metode standarisasi yang umum digunakan adalah *StandardScaler*, yang mengubah distribusi setiap fitur numerik agar memiliki rata-rata nol dan deviasi standar satu. (Thakker & Buch, 2024) menemukan bahwa pemilihan teknik scaling yang tepat sangat berpengaruh terhadap performa model regresi. Penting untuk diperhatikan bahwa proses scaling hanya diterapkan berdasarkan statistik data latih, kemudian transformasi yang sama diterapkan pada data uji untuk menghindari data leakage.

4. Transformasi Logaritma pada Variabel Target

Transformasi logaritma pada variabel target umum digunakan dalam pemodelan harga properti untuk menstabilkan distribusi data yang cenderung bersifat *right-skewed* akibat rentang harga yang sangat lebar. Salah satu bentuk penerapannya adalah model harga hedonik dalam bentuk semi-logaritmik, di mana distribusi harga rumah disesuaikan ke skala logaritma sebelum dilakukan pemodelan regresi (Wu et al., 2021). Bentuk semi-logaritmik ini dipilih karena dapat secara efektif mengatasi atau mengurangi masalah heteroskedastisitas pada model regresi, dengan bentuk persamaan yang relatif sederhana (Wu et al., 2021). Dengan demikian, transformasi logaritma pada variabel target tidak hanya berfungsi menstabilkan varians, tetapi juga membantu model regresi memenuhi asumsi-asumsi statistik yang mendasarinya.

5. Perbedaan antara *Data cleaning* dan *Preprocessing*

Perbedaan antara *Data cleaning* dan *preprocessing* terletak pada ruang lingkup dan tujuan utama masing-masing. *Data cleaning* berfokus pada identifikasi dan penanganan data yang hilang, duplikat, atau anomali seperti *outlier*, sedangkan *Data Preprocessing* mencakup tahap-tahap lebih luas yang meliputi transformasi data, *encoding*, *split data*, dan *scaling*. Menurut (Thakker & Buch, 2024), proses pembersihan data merupakan bagian awal dari *preprocessing* yang penting untuk menjamin integritas data sebelum diterapkan teknik transformasi dan pelatihan model.

2.2.12 Algoritma Regresi yang Digunakan

2.2.12.1. *Regresi Linear*

Regresi Linear merupakan salah satu metode statistik paling dasar namun efektif dalam menganalisis hubungan antara satu variabel dependen (respon) dengan satu atau lebih variabel independen (prediktor). Model ini digunakan secara luas dalam berbagai bidang, termasuk prediksi harga rumah, karena kesederhanaannya, interpretabilitas yang tinggi, dan kemudahan implementasinya. Sebagai model yang tergolong sederhana, *Regresi Linear* memiliki sejumlah kelebihan yang menjadikannya pilihan relevan dalam berbagai kasus prediksi. Koefisien regresi yang dihasilkan dapat langsung diinterpretasikan sebagai kontribusi masing-masing variabel terhadap harga, sehingga memudahkan pemahaman hubungan antara fitur dan target. Selain itu, proses pelatihan model ini secara komputasi sangat efisien dan tidak memerlukan *tuning hyperparameter* yang kompleks, sehingga cocok digunakan sebagai model dasar maupun pembanding dalam sistem prediksi berbasis machine learning (Nuris, 2024). Dalam konteks prediksi harga rumah, *Regresi Linear* digunakan untuk mengestimasi harga berdasarkan fitur-fitur seperti luas bangunan, jumlah kamar, kualitas bangunan, dan variabel lainnya.

(Sharma et al., 2024) dalam penelitiannya menggunakan *Regresi Linear* sebagai salah satu metode pembanding dalam prediksi harga rumah menggunakan *dataset Ames Housing*. *Regresi Linear* dikembangkan sebagai model dasar yang membentuk hubungan antara variabel target (harga rumah) dengan variabel-variabel prediktor. Hubungan ini secara matematis diformulasikan seperti pada persamaan (2.1):

$$Y = \beta_0 + \sum_{i=1}^k \beta_i X_i + \varepsilon \quad (2.1)$$

Di mana:

- Y adalah variabel dependen (harga rumah),
- β_0 adalah intersep (konstanta),

- β_i adalah koefisien regresi dari masing-masing variabel independen X_i ,
- X_i adalah variabel prediktor (fitur rumah),
- ϵ adalah *error* atau kesalahan residual.

Dalam penelitian tersebut, meskipun *Regresi Linear* menunjukkan performa yang cukup baik, dengan nilai *R-squared* sebesar 0,935 dan nilai *Mean Absolute Error* (MAE) terendah dibandingkan model lainnya yaitu sebesar 0.075, namun model ini masih kalah dari *XGBoost* yang memberikan performa prediksi lebih tinggi secara keseluruhan (Sharma et al., 2024). Hal ini menunjukkan bahwa *Regresi Linear* cocok untuk digunakan sebagai *baseline* model, tetapi kurang optimal dalam menangkap kompleksitas hubungan non-linear antara fitur-fitur dan harga rumah yang sebenarnya.

Meskipun demikian, *Regresi Linear* memiliki sejumlah keterbatasan yang perlu diperhatikan. Model ini mengasumsikan adanya hubungan linear yang ketat antara variabel independen dan variabel dependen, sehingga performanya cenderung menurun ketika data mengandung pola yang lebih kompleks atau non-linear. Selain itu, *Regresi Linear* cukup sensitif terhadap keberadaan *outlier*, di mana nilai ekstrem pada data dapat memengaruhi koefisien model secara signifikan dan mengurangi akurasi prediksi secara keseluruhan. Model ini juga rentan terhadap masalah multikolinearitas, yaitu kondisi di mana terdapat korelasi tinggi antara dua atau lebih variabel prediktor, yang dapat menyebabkan estimasi koefisien menjadi tidak stabil dan sulit diinterpretasikan (Sharma et al., 2024). Oleh karena itu, meskipun *Regresi Linear* layak digunakan dalam tahap awal eksplorasi data atau sebagai model pembanding, penggunaan model yang lebih kompleks seperti metode *Ensemble* dapat meningkatkan akurasi prediksi secara signifikan.

2.2.12.2. *Random Forest*

Random Forest merupakan algoritma *machine learning* berbasis *ensemble learning* yang dibangun dari penggabungan beberapa pohon

keputusan (*decision tree*). Metode ini dikembangkan dari teknik *Classification and Regression Tree (CART)* dengan pendekatan *bootstrap aggregating* atau *bagging* serta pemilihan fitur secara acak (*random feature selection*). Prinsip dasarnya adalah membangun sejumlah pohon keputusan secara paralel, kemudian menggabungkan hasil prediksi dari masing-masing pohon untuk menghasilkan *output* akhir berdasarkan *voting* atau rata-rata (Hadi & Benedict, 2024).

Dalam konteks regresi, *Random Forest* berperan untuk memperkirakan nilai kontinu seperti harga rumah. Setiap pohon regresi memberikan prediksi, lalu hasil dari seluruh pohon akan dirata-ratakan untuk memperoleh nilai akhir. Salah satu keunggulan *Random Forest* adalah kemampuannya dalam menangani data dengan dimensi tinggi, serta ketahanannya terhadap *overfitting* dibandingkan dengan *single decision tree*.

Di sisi lain, *Random Forest* juga memiliki beberapa keterbatasan yang perlu dipertimbangkan. Karena membangun sejumlah besar pohon keputusan secara bersamaan, model ini membutuhkan waktu pelatihan dan sumber daya memori yang lebih besar dibandingkan model tunggal, terutama ketika jumlah estimator dan dimensi fitur yang digunakan cukup tinggi. Selain itu, *Random Forest* tergolong sebagai model black-box yang sulit diinterpretasikan secara langsung, sehingga tidak mudah untuk menjelaskan bagaimana keputusan prediksi dihasilkan dibandingkan dengan model seperti Regresi Linear (Fauzi et al., 2025; Hadi & Benedict, 2024)

Secara umum, pemilihan fitur dan pembentukan node dalam *Random Forest* didasarkan pada pengukuran *entropy* dan *information gain*. Nilai *entropy* dihitung untuk mengetahui tingkat ketidakmurnian (*impurity*) dari atribut yang digunakan, sebagaimana ditunjukkan pada Persamaan (2.2).

$$Entropy(Y) = - \sum P(c | Y) \cdot \log_2 P(c | Y) \quad (2.2)$$

Keterangan:

- Y adalah himpunan kasus,
- $P(c | Y)$ adalah proporsi kelas c dalam himpunan Y .

Kemudian, nilai *Information Gain* dihitung untuk menentukan atribut terbaik dalam pemisahan data, sebagaimana ditunjukkan pada Persamaan (2.3):

$$Information\ Gain(Y, a) = Entropy(Y) - \sum_{v \in Values(a)} \frac{|Y_v|}{|Y_a|} \cdot Entropy(Y_v) \quad (2.3)$$

Keterangan:

- Y_v adalah subset dari Y dengan nilai atribut a tertentu,
- Y_a adalah keseluruhan data untuk atribut a .

Penelitian oleh (Hadi & Benedict, 2024) menerapkan algoritma *Random Forest* untuk prediksi harga rumah menggunakan *dataset King County, USA* yang diperoleh dari *Kaggle*. *Dataset* tersebut mencakup 21.613 data dengan 13 variabel prediktor, antara lain luas rumah, jumlah kamar, grade bangunan, dan lainnya. Setelah dilakukan proses *Data Preprocessing* dan pemisahan data (70% *training*, 30% *testing*), model dievaluasi menggunakan metrik *R-squared* (R^2) dan *Root Mean Square Error* (RMSE).

Hasil pengujian menunjukkan bahwa model *Random Forest* dengan jumlah pohon sebanyak 20 menghasilkan performa terbaik, dengan nilai R^2 sebesar 0,8654 dan RMSE sebesar 144.913,73. Hasil ini lebih unggul dibandingkan dua algoritma pembanding lainnya, yakni *Decision Tree* dengan R^2 sebesar 0,7639 dan *Polynomial Regression* dengan R^2 sebesar 0,7813 (Hadi & Benedict, 2024).

2.2.12.3. *XGBoost*

Extreme Gradient Boosting (XGBoost) adalah salah satu algoritma *ensemble learning* berbasis *boosting* yang dirancang untuk mengatasi berbagai permasalahan prediksi, baik klasifikasi maupun regresi, secara

cepat dan akurat. *XGBoost* bekerja dengan cara membangun pohon keputusan secara bertahap, di mana setiap pohon baru dibentuk untuk memperbaiki kesalahan prediksi dari pohon sebelumnya. Proses ini membuat *XGBoost* sangat efektif dalam menangkap pola *non-linear* dan kompleks dalam data (Rayadin et al., 2024).

Algoritma ini dikembangkan dengan fokus pada efisiensi komputasi dan akurasi prediksi. *XGBoost* mendukung proses pelatihan paralel, teknik pruning untuk menghindari *overfitting*, serta penanganan terhadap nilai hilang (*missing value*). Selain itu, *XGBoost* juga memiliki mekanisme regularisasi yang secara eksplisit mengontrol kompleksitas model sehingga mampu menghasilkan model yang lebih general dan stabil (Rayadin et al., 2024).

Secara matematis, prediksi akhir dari model *XGBoost* didefinisikan sebagai penjumlahan dari sejumlah fungsi pohon keputusan, sebagaimana ditunjukkan pada persamaan (2.4):

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} \quad (2.4)$$

dengan $\mathcal{F}$ merupakan himpunan dari semua pohon regresi yang mungkin terbentuk. Fungsi objektif yang dioptimasi oleh *XGBoost* terdiri dari dua komponen utama, yaitu fungsi *loss* dan fungsi regularisasi, sebagaimana ditunjukkan pada persamaan (2.5):

$$\mathcal{L} = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2.5)$$

Adapun fungsi regularisasi $\Omega(f)$ bertujuan untuk mengontrol kompleksitas model dan didefinisikan sebagaimana pada persamaan (2.6):

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (2.6)$$

dengan:

- T adalah jumlah daun pada pohon,

- w adalah bobot pada setiap daun,
- γ adalah parameter kompleksitas pohon,
- λ adalah parameter regularisasi L2 (Rayadin et al., 2024).

Dalam konteks prediksi harga properti, *XGBoost* secara konsisten menunjukkan performa yang unggul. Sharma et al. (2024) melaporkan bahwa *XGBoost* mencapai nilai R^2 sebesar 0,997 pada *dataset* Ames Housing setelah dilakukan *tuning hyperparameter*, menjadikannya model terbaik dibandingkan Support Vector Regressor, *Random Forest*, Multilayer Perceptron, dan *Multiple Linear Regression*. Temuan serupa dilaporkan oleh Ade & Sherifdeen (2024), di mana *XGBoost* menunjukkan akurasi dan ketahanan tertinggi pada *dataset* besar dengan banyak fitur, melampaui *Random Forest* dan *Linear Regression* dalam hal presisi prediksi harga rumah. Lebih lanjut, Aqsha (2025) dalam penelitiannya yang secara khusus membandingkan *XGBoost* dan *Random Forest* untuk prediksi harga rumah di Jabodetabek juga menegaskan keunggulan *XGBoost* sebagai algoritma yang lebih optimal untuk kasus serupa di Indonesia.

Meskipun demikian, *XGBoost* memiliki beberapa keterbatasan yang perlu diperhatikan. Model ini memerlukan proses *tuning hyperparameter* yang lebih intensif dibandingkan *Random Forest*, karena parameter seperti *n_estimators*, *learning_rate*, dan *max_depth* harus dikonfigurasi secara cermat agar model tidak mengalami *overfitting* maupun *underfitting*. Selain itu, seperti halnya *Random Forest*, *XGBoost* juga tergolong model *black-box* yang tingkat interpretabilitasnya lebih rendah dibandingkan Regresi Linear, sehingga kurang ideal digunakan secara mandiri pada kasus yang mengutamakan transparansi model (Sharma et al., 2024; Aqsha, 2025).

Pemilihan *Linear Regression*, *Random Forest*, dan *XGBoost* sebagai *base learner* dalam penelitian ini didasarkan pada prinsip diversitas model, yaitu bahwa kombinasi algoritma dengan karakteristik yang saling melengkapi cenderung menghasilkan prediksi yang lebih akurat dan stabil dibandingkan model tunggal (Hibatulloh et al., 2025; Liang et al., 2021).

Linear Regression memberikan kontribusi berupa interpretabilitas tinggi dan kemampuan menangkap hubungan linear dalam data, *Random Forest* berkontribusi dalam menangani hubungan non-linear dan interaksi kompleks antarvariabel, sementara *XGBoost* melengkapi keduanya dengan efisiensi komputasi dan performa prediksi yang secara konsisten unggul pada berbagai kasus prediksi harga properti (Ade & Sherifdeen, 2024; Sharma et al., 2024). Kombinasi ketiganya memastikan bahwa *meta learner* pada *Stacking Regressor* menerima sinyal prediksi yang bervariasi dari perspektif yang berbeda, sehingga kelemahan satu algoritma dapat dikompensasi oleh kekuatan algoritma lainnya.

2.2.12.4. *Stacking Regressor*

Stacking, atau *stacked generalization*, adalah metode *ensemble learning* yang menggabungkan prediksi dari beberapa model pembelajaran mesin (disebut *base learners* atau model tingkat-0) melalui sebuah model pelajar kedua (*meta learner* atau model tingkat-1). Tujuan utama metode ini adalah untuk meningkatkan performa prediksi dengan memanfaatkan kekuatan relatif dari masing-masing algoritma dasar yang digunakan (Liang et al., 2021; Zhang et al., 2022).

Secara umum, struktur *stacking* terdiri dari dua lapisan. Pada lapisan pertama, sejumlah model dasar dilatih menggunakan data pelatihan asli. Kemudian, prediksi dari model-model tersebut digunakan sebagai fitur masukan untuk melatih *meta-model*, yang bertugas menggabungkan hasil prediksi tersebut secara optimal (Hibatulloh et al., 2025). Formulasi matematis dari proses ini dapat dijelaskan sebagaimana pada persamaan (2.7):

Diberikan *dataset* pelatihan (X, y) , *Stacking* terdiri atas:

1. *Base learners*: $f_1(X), f_2(X), \dots, f_k(X)$
2. *Meta-model*: f_{meta}

Output stacking:

$$\hat{y} = f_{meta}(f_1(X), f_2(X), \dots, f_k(X)) \quad (2.7)$$

Keterangan:

- X : matriks fitur dari data pelatihan
- $f_1, f_2, \dots, f_k$: model-model dasar (*base learners*)
- f_{meta} : model meta (*meta learner*), dilatih pada *output* dari *base learners*
- $\hat{y}$: prediksi akhir dari model *stacking*

Untuk memastikan bahwa setiap *base learner* dievaluasi secara internal sebelum prediksi akhirnya dikombinasikan oleh *meta learner*, mekanisme out-of-fold (OOF) diterapkan dalam proses pelatihan *Stacking Ensemble*. Dalam implementasinya, terdapat dua varian pendekatan untuk menghasilkan input bagi *meta learner*. Pada varian OOF klasik, *base learner* dilatih menggunakan K-Fold pada data pelatihan dan prediksi pada *fold* validasi dikumpulkan sebagai meta-features untuk melatih *meta learner*. Pada varian OOF klasik yang diterapkan dalam penelitian ini, data *training* dibagi menjadi K *fold*. Pada setiap iterasi, *base learner* dilatih dari K-1 *fold* dan menghasilkan prediksi pada satu *fold* yang belum pernah dilihat. Setelah seluruh iterasi selesai, kumpulan prediksi OOF dari seluruh *fold* digabungkan membentuk matriks meta-features yang bersifat jujur karena setiap prediksi dihasilkan oleh model yang tidak pernah melihat data tersebut saat pelatihan. Matriks inilah yang digunakan sebagai input untuk melatih *meta learner*, sehingga tidak diperlukan set data terpisah. Setelah *meta learner* selesai dilatih, *base learner* di-retrain menggunakan keseluruhan data *training* sebelum digunakan untuk menghasilkan prediksi pada data test (Park et al., 2022).

Berbeda dari metode *voting*, yang hanya mengambil rata-rata (regresi) atau modus (klasifikasi) dari prediksi semua model dasar, *stacking* secara eksplisit mempelajari cara menggabungkan prediksi tersebut. Sementara *bagging* (misalnya *Random Forest*) melatih beberapa model pada subset data yang berbeda dan menggabungkannya dengan agregasi sederhana untuk mengurangi varians, *Stacking* menggunakan pendekatan

pembelajaran terarah untuk menyatukan prediksi (Hibatulloh et al., 2025; Liang et al., 2021).

Dalam konteks prediksi harga properti, pendekatan *Stacking Regressor* menunjukkan performa yang lebih baik dibandingkan model tunggal maupun metode *Ensemble* lainnya. (Hibatulloh et al., 2025) melaporkan bahwa *Stacking Ensemble* memperoleh kinerja terbaik dalam memprediksi harga rumah di Bandung dengan nilai R^2 sebesar 0,9076, RMSE sebesar 0,311, dan MAE sebesar 0,216, melampaui *Random Forest*, *Gradient Boosting*, *XGBoost*, dan *Linear Regression* yang diuji secara individual. Hal ini menunjukkan bahwa kemampuan *meta learner* dalam mempelajari kombinasi optimal dari prediksi *base learner* mampu mengkompensasi kelemahan masing-masing algoritma dasar secara efektif.

Keunggulan utama *Stacking Regressor* terletak pada fleksibilitas dan kemampuan generalisasinya yang tinggi. Dengan menggabungkan model-model yang memiliki karakteristik berbeda, *Stacking* dapat menangkap pola dalam data secara lebih menyeluruh dibandingkan model tunggal. Pendekatan ini juga tidak bergantung pada asumsi bahwa satu algoritma tertentu selalu lebih baik, melainkan membiarkan *meta learner* menentukan kontribusi optimal dari setiap *base learner* secara adaptif (Liang et al., 2021; Zhang et al., 2022). Namun demikian, *Stacking Regressor* juga memiliki keterbatasan. Proses pelatihannya lebih kompleks dan membutuhkan waktu komputasi yang lebih lama karena melibatkan pelatihan beberapa model secara bertahap. Selain itu, pemisahan set data yang ketat antar setiap tahap pelatihan perlu dilakukan secara cermat untuk mencegah terjadinya data leakage yang dapat mengakibatkan estimasi performa model menjadi tidak realistis (Hibatulloh et al., 2025; Zhang et al., 2022).

Dalam praktiknya, berbagai algoritma dapat digunakan sebagai *base learners*, seperti *Random Forest*, *Support Vector Regression (SVR)*, *Gradient Boosting*, *Multilayer Perceptron (MLP)*, dan *Elastic Net*. Meta-model yang umum digunakan meliputi regresi linier (OLS, ridge, lasso), *Random Forest*, atau bahkan neural networks ringan, tergantung pada

kompleksitas data dan tujuan model (Liang et al., 2021). Di antara pilihan tersebut, Ridge Regression merupakan salah satu *meta learner* yang banyak digunakan dalam konteks *Stacking* karena kemampuan regularisasinya dalam mengurangi *overfitting*, terutama ketika prediksi dari beberapa *base learner* berkorelasi satu sama lain. Dengan menambahkan penalti terhadap besarnya koefisien, Ridge Regression mampu menghasilkan kombinasi prediksi yang lebih stabil dan general dibandingkan regresi linier biasa (Liang et al., 2021). Oleh karena itu, parameter alpha pada Ridge Regression perlu dikonfigurasi secara optimal melalui proses *hyperparameter tuning*, mengingat nilai alpha menentukan seberapa kuat regularisasi diterapkan terhadap kombinasi prediksi dari *base learner* yang berpotensi saling berkorelasi, sehingga pemilihan nilai alpha yang tepat menjadi krusial untuk memastikan *meta learner* menghasilkan prediksi akhir yang akurat dan stabil (Nascimento et al., 2024).

2.2.13 *Hyperparameter Tuning*

Dalam pengembangan model *machine learning*, terdapat dua jenis parameter yang perlu dipahami secara konseptual, yaitu parameter model dan *hyperparameter*. Parameter model merupakan nilai yang diperoleh secara otomatis melalui proses pelatihan dari data, sedangkan *hyperparameter* merupakan konfigurasi atau pengaturan yang harus ditentukan secara manual oleh peneliti sebelum proses pelatihan dimulai (Arifin et al., 2023). *Hyperparameter* berperan dalam mengontrol struktur dan perilaku model secara keseluruhan, sehingga kombinasi nilai *hyperparameter* yang dipilih dapat berdampak signifikan terhadap kinerja prediksi yang dihasilkan. Oleh karena itu, proses pencarian kombinasi *hyperparameter* yang optimal, yang dikenal sebagai *hyperparameter tuning* atau *hyperparameter optimization*, menjadi tahapan penting dalam membangun model *machine learning* yang akurat dan andal (Zhao et al., 2024).

Salah satu metode yang paling umum digunakan untuk melakukan *hyperparameter tuning* adalah *Grid Search*. *Grid Search* merupakan teknik optimasi yang bekerja dengan cara menelusuri seluruh kemungkinan

kombinasi nilai *hyperparameter* yang telah didefinisikan oleh pengguna dalam suatu ruang pencarian (*search space*) (Arifin et al., 2023). Setiap kombinasi *hyperparameter* dievaluasi secara sistematis, dan kombinasi yang menghasilkan nilai metrik evaluasi terbaik dipilih sebagai konfigurasi optimal. Meskipun pendekatan ini bersifat menyeluruh dan menjamin bahwa seluruh kombinasi yang didefinisikan telah diperiksa, metode ini memiliki konsekuensi berupa meningkatnya waktu komputasi secara signifikan seiring bertambahnya jumlah *hyperparameter* dan rentang nilainya (Zhao et al., 2024). Meskipun demikian, *Grid Search* tetap menjadi pilihan yang direkomendasikan ketika jumlah *hyperparameter* yang dioptimasi relatif terbatas, karena kesederhanaannya dan kemampuannya yang telah terbukti dalam meningkatkan akurasi model (Malakouti et al., 2023).

Dalam praktiknya, *Grid search* umumnya dikombinasikan dengan teknik validasi silang *K-Fold Cross Validation* menjadi *GridSearchCV*, sehingga evaluasi terhadap setiap kombinasi *hyperparameter* dilakukan secara lebih *robust* dan tidak bergantung pada satu pembagian data tertentu (Malakouti et al., 2023). Pendekatan ini disebut *flat cross-validation*, karena satu tahap validasi silang digunakan sekaligus untuk memilih *hyperparameter* terbaik dan mengestimasi performa model. Pendekatan ini berbeda dari *nested cross-validation*, yang memisahkan pencarian *hyperparameter* (melalui *inner loop*) dari estimasi performa yang tidak bias (melalui *outer loop*), sehingga lebih ketat secara statistik namun membutuhkan biaya komputasi yang jauh lebih tinggi. Studi yang membandingkan kedua pendekatan tersebut pada dua belas algoritma dan 115 *dataset* menemukan bahwa *flat cross-validation* yang secara komputasi jauh lebih murah umumnya tidak menghasilkan pemilihan algoritma yang lebih buruk dibandingkan *nested cross-validation*, khususnya pada algoritma seperti *Random Forest* dan *gradient boosting machine* (Yu et al., 2024). Kombinasi *Grid search* dengan *K-Fold Cross Validation* dalam *GridSearchCV* terbukti efektif dalam menekan risiko *overfitting* sekaligus

memastikan *hyperparameter* yang terpilih menghasilkan model dengan kemampuan generalisasi yang baik (Arifin et al., 2023).

2.2.14 Evaluasi Model Regresi

Evaluasi model merupakan tahap krusial dalam proses pengembangan *machine learning*, karena menentukan sejauh mana model yang dibangun mampu memberikan hasil prediksi yang akurat dan dapat diandalkan. Evaluasi dilakukan dengan membandingkan hasil prediksi model dengan nilai aktual (*ground truth*) pada data uji (*test data*) yang belum pernah dilihat model sebelumnya. Melalui evaluasi, peneliti dapat menilai performa model, mendeteksi potensi *overfitting* atau *underfitting*, serta memilih model terbaik di antara beberapa alternatif yang diuji (Erdebilli & Devrim-İçtenbaş, 2022).

Dalam permasalahan regresi, performa model umumnya diukur menggunakan beberapa metrik utama, yaitu *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), dan *R-squared* (koefisien determinasi).

1. *Mean Absolute Error (MAE)*

MAE mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual. Metrik ini menunjukkan seberapa jauh hasil prediksi rata-rata menyimpang dari nilai sebenarnya. Perhitungan MAE secara matematis dapat dinyatakan sebagaimana pada persamaan (2.8):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.8)$$

2. *Root Mean Squared Error (RMSE)*

RMSE mengukur akar dari rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual. Nilai RMSE yang lebih kecil menunjukkan kesalahan prediksi yang lebih rendah, tetapi metrik ini lebih sensitif terhadap *outlier* dibanding MAE. Secara matematis, nilai RMSE dapat dihitung menggunakan formulasi pada persamaan (2.9):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.9)$$

3. *Mean Absolute Percentage Error (MAPE)*

MAPE mengukur kesalahan prediksi dalam bentuk persentase relatif terhadap nilai aktual. Metrik ini mempermudah interpretasi kesalahan karena dinyatakan dalam satuan persentase. Secara matematis, MAPE dihitung menggunakan formulasi pada persamaan (2.10):

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.10)$$

4. *R-squared (Koefisien Determinasi)*

R-squared menunjukkan proporsi variasi pada variabel dependen yang dapat dijelaskan oleh model. Nilai R-squared berkisar antara 0 hingga 1; semakin mendekati 1, semakin baik model dalam menjelaskan data. Secara matematis, nilai R-squared dapat dihitung menggunakan formulasi pada persamaan (2.11):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.11)$$

Di mana:

- y_i adalah nilai aktual,
- $\hat{y}_i$ adalah nilai prediksi,
- $\bar{y}$ adalah nilai rata-rata aktual,
- n adalah jumlah sampel data.

Dalam studi kasus yang dilakukan oleh (Erdebilli & Devrim-İçtenbaş, 2022), metrik-metrik tersebut digunakan untuk mengevaluasi kinerja *Random Forest*, *Gradient Boosting Machine*, *AdaBoost*, dan model *Ensemble Voting Regressor* (VR). Hasil evaluasi menunjukkan bahwa model VR yang menggabungkan beberapa algoritma mampu menghasilkan

nilai RMSE dan MAE yang lebih rendah serta nilai *R-squared* yang lebih tinggi dibandingkan model tunggal. Hal ini menunjukkan bahwa model *ensemble* memiliki keunggulan dalam meminimalkan kesalahan prediksi dan meningkatkan stabilitas performa.

2.2.15 Validasi Model (*Cross Validation*)

Cross-validation merupakan salah satu teknik evaluasi model yang umum digunakan dalam pembelajaran mesin untuk mengukur kemampuan generalisasi model terhadap data yang tidak pernah dilihat sebelumnya. Teknik ini juga berfungsi mengurangi risiko *overfitting* dan bias evaluasi (Moscovich & Rosset, 2022). Di antara berbagai pendekatan yang tersedia, *K-fold cross-validation* menjadi metode yang paling banyak diadopsi dalam berbagai penelitian.

Pada *K-fold cross-validation*, data pelatihan dibagi secara merata menjadi *K subset* yang disebut *fold*. Proses pelatihan dan validasi dilakukan sebanyak *K iterasi*. Pada setiap iterasi *iii*, satu *fold* digunakan sebagai data validasi $D_{val}^{(i)}$ dan $K - 1$ *fold* sisanya digunakan sebagai data pelatihan $D_{train}^{(i)}$. Nilai metrik evaluasi, misalnya kesalahan prediksi E_i , dihitung untuk masing-masing iterasi, lalu rata-ratanya digunakan sebagai estimasi performa akhir model. Rumus umum *K-fold cross-validation* dapat diformulasikan sebagaimana pada persamaan (2.12).

$$E_{CV} = \frac{1}{K} \sum_{i=1}^K E_i \quad (2.12)$$

dengan:

- E_{CV} = estimasi kesalahan rata-rata model,
- E_i = kesalahan pada *fold* ke- *i*,
- K = jumlah *fold*.

Tujuan utama metode ini adalah menurunkan bias dan varians estimasi performa model. Dibandingkan pembagian data tunggal (*hold-out*

validation) yang cenderung menghasilkan estimasi bias, atau *leave-one-out cross-validation* yang memiliki varians lebih besar dan kompleksitas komputasi tinggi, *K-fold cross-validation* menawarkan keseimbangan yang baik antara akurasi estimasi dan efisiensi komputasi (Nti et al., 2021).

Namun, penting untuk dicatat bahwa proses *preprocessing* yang bersifat tidak terawasi (*unsupervised preprocessing*), seperti normalisasi, standarisasi, atau seleksi fitur, harus dilakukan hanya pada data pelatihan di setiap iterasi, kemudian diterapkan pada data validasi. Jika langkah ini dilakukan pada seluruh data sebelum pembagian *fold*, akan terjadi kebocoran informasi (*data leakage*) yang dapat mengakibatkan bias positif pada hasil evaluasi (Moscovich & Rosset, 2022).

Dalam praktiknya, tidak terdapat aturan baku mengenai nilai K yang harus digunakan. Nilai $K=5$ dan $K=10$ merupakan nilai yang paling umum digunakan dalam literatur, karena keduanya diyakini memberikan estimasi kesalahan prediksi yang tidak terlalu bias maupun memiliki varians yang terlalu tinggi (Nti et al., 2021). Namun demikian, penelitian yang membandingkan performa berbagai nilai K secara empiris menemukan bahwa nilai K yang lebih kecil, seperti $K=5$, lebih disarankan untuk *dataset* besar dengan algoritma yang memiliki kompleksitas komputasi tinggi, guna mencegah peningkatan waktu komputasi yang berlebihan (Nti et al., 2021).

Sebagai perluasan dari metode ini, *Repeated K-Fold Cross-validation* menjalankan seluruh proses *K-fold* sebanyak beberapa kali dengan pembagian data yang berbeda pada setiap pengulangannya, sehingga estimasi performa akhir diperoleh dari rata-rata seluruh hasil *fold* di semua pengulangan. Pendekatan ini terbukti menghasilkan estimasi yang lebih stabil sekaligus memungkinkan pengukuran variabilitas model melalui nilai standar deviasi metrik evaluasi, yang tidak dapat diperoleh dari satu putaran *K-fold* biasa (Lumumba et al., 2024).

2.2.16 Antarmuka Pengujian Model

Antarmuka pengujian model adalah suatu tampilan interaktif yang digunakan untuk menguji model pembelajaran mesin dengan memasukkan

data masukan (*input*) secara langsung melalui komponen visual seperti formulir, *dropdown menu*, atau tombol eksekusi. Antarmuka ini mempermudah proses interaksi antara pengguna dan model yang telah dibangun, sehingga pengguna tidak perlu menulis kode program untuk melakukan prediksi. Melalui antarmuka, pengguna dapat mengubah nilai parameter fitur dan langsung melihat dampaknya terhadap hasil prediksi secara real-time. Pendekatan ini sangat bermanfaat dalam tahap evaluasi model karena dapat mempercepat pengujian serta meningkatkan aksesibilitas model bagi pengguna non-teknis.

Pengembangan antarmuka pengujian model pada penelitian ini akan menggunakan *Streamlit*, yaitu sebuah *open-source* Python framework yang dirancang khusus untuk membangun aplikasi web interaktif berbasis data secara cepat tanpa memerlukan keahlian pengembangan web yang kompleks. *Streamlit* mendukung integrasi langsung dengan model pembelajaran mesin dan menyediakan berbagai komponen interaktif seperti formulir input, grafik, dan visualisasi data dalam satu halaman aplikasi. Menurut (Tegar et al., 2025), *Streamlit* dipilih karena kemampuannya dalam menyederhanakan proses pembuatan antarmuka, sehingga pengguna non-teknis dapat menjalankan model dan melihat hasil prediksi secara langsung. Dalam penelitiannya, *Streamlit* digunakan untuk mengembangkan aplikasi prediksi yang memungkinkan pengguna memasukkan parameter masukan, menjalankan model, serta menampilkan hasil prediksi dan visualisasi pendukung secara real-time, sehingga proses validasi model menjadi lebih cepat, intuitif, dan mudah diakses.